DON'T BREAK THE RULES OR HELPING NON-CARTOGRAPHERS TO DESIGN MAPS: AN APPLICATION FOR CARTOGRAPHIC EXPERT SYSTEMS.

David Forrest
Department of Geography & Topographic Science
University of Glasgow
GLASGOW G12 8QQ
United Kingdom
Fax: 041 330 4894

ABSTRACT

There are now many systems available allowing non cartographically trained users to produce maps. In order to help these users it is essential that cartographic knowledge be incorporated into the systems they are using. MapDesigner is a knowledge based system for producing small scale maps from national or regional databases. It has a built in knowledge base containing rules for the selection and representation of information for small scale regional maps and relies heavily on meta data relating to the contents on the spatial and attribute databases used to generate the maps.The system is written in Prolog and makes extensive use of that language's capabilities for list processing and backtracking to find solutions. The primary knowledge structure is the frame. For a map to be output each slot in the frame must be completed. Whenever possible, menus, etc. are generated at run time from the knowledge base, allowing the system to respond to the current situation and to allow for easy expansion of the knowledge base.

INTRODUCTION

The continual drop in price and enhancement in performance of computer hardware has brought the possibility of computer mapping to a wider audience. This increase in the availability of computer mapping capabilities has lead to a rise in the number of potential map authors, but does not however appear to have lead to more widespread knowledge of cartographic design principles. As it is unlikely that the general level of cartographic education of most computer map authors will be greatly increased, cartographers must strive to make the programs used by naive map authors better able to produce reasonably well designed maps, or at least maps which do not break the most fundamental rules of map design. To enable such a system to be developed the basic map design process must be formalised.

The research reported on here emphasises this formalisation of the map design process, both in terms of the major processes involved and in terms of a formal structure for and definition of the various representation methods which can be used for visualisation of spatial data. This formalisation is used as the basis of an expert system for designing small scale maps such as may be found in regional atlases. The expert system uses a frame based approach for knowledge representation which was found to be well suited to the structure of map definition. Rather than concentrating on one limited aspect of map design the approach taken is to model the whole map making process for a wide range of map types, but over a limited range of scales, thus limiting several problems of generalisation. The systems considers not only the specific subject information for the required map theme, but also the selection and representation of the appropriate base information. This results in the need to resolve complex interactions and to determine appropriate visual hierarchies.

One specific aspect that becomes clear in developing such a system is that not only is cartographic knowledge required, but that there is a requirement for adequate knowledge about the characteristics of the information being mapped. The nature of this meta data is also reported on.

THE MAP DESIGN PROCESS

The general procedure for designing a map follows the route of Compose; Compile; Symbol Specification; Display; Adjust. Logically an essentially similar route should be followed by a Cartographic Design Expert System. The main modules of the MapDesiger system are Description; Layout; Data Selection; Symbolisation; Display and Modify.
description

Description

The first module requires the user to describe to the system some of the basic characteristics of the required map. This includes information about intended map use and map user together with the output medium (e.g. screen , pen plot, slide, etc.) This is then followed by selecting the main topic of the map. The system will 'know' about a definitive range of maps and the aim at this stage is to get the system user to choose the type of map that most closely matches his requirement. As discussed below, the map topics are defined hierarchically so it is always possible to select a map topic.

Layout

The layout module determines the area to be mapped, the scale and the format. Clearly if the first and either of the other two are selected the third can be determined by the system. Checks are carried out to ensure that valid parameters are set.

Selection

The information to be included in a map, both base and topic, will depend upon the map topic, the scale and the level of detail required. A selection index is computed by the system based upon the information gathered previously and is used in conjunction with a series of facts in the knowledge base to determine what base and topic information should be included. Thus, the base map information included will be appropriate to the map topic selected. Non relevant information will be excluded. The level of rivers or roads for example included as base information will also depend upon the scale of the map and the level of detail required, the level of detail being determined with reference to the intended use, the user and the output medium. It is important to note here that the concept of the standard topographic base does not exist in the system. The base information is selected specifically for each map.

Symbol Specification

The first stage in specifying the symbols is to assign an appropriate cartographic representation method to each class of information to be included in the map. Some classes of information only have one sensible representation method, but for some data there are alternative, for example relief may be shown by contours or hypsometric colours. The system ensures that there are no conflicting representations chosen, but that as far as possible each class of information is assigned the most desirable representation method.

Having assigned the representation method, individual symbols are assigned to each feature within each class of information. This is the largest set of rules and is the most problematic area of developing a system due to the almost limitless number of permutations possible, but particularly in relation to the problem of checking for conflicts between symbols.

Display

Having determined the information to be included and its graphical representation, the map can be displayed on the screen. Generally this is a straight forward procedural operation, but the system does incorporate some novel elements. For example the colour assigned to areas for a politically coloured map are determined by an expert system component. This ensures that no two adjacent zones receive the same colour, i.e. the system can resolve the standard four colour map problem.

Modify

It would be ambitious to assume that the system could always design the ideal map at the first attempt, therefore an ability to interact with the user to modify the map is an essential element. To date this part of the system has not been tackled. As there is no satisfactory way to analytically assess map design, some form of interactive assessment will be necessary. Problems of how to describe the changes required will have to be resolved, particularly as the target users of the system are not cartographers.

THE SYSTEM

The MapDesigner system essentially consists of four parts. These are the database of geographic information; the knowledge base which holds information on how to use the data; a user interface; and a control program (inference engine) which interacts with the user interface, the database and the knowledge base and guides inference. In addition to these primary components, several supporting elements are also required for a complete cartographic design expert system. As shown in Figure 1, these include a knowledge acquisition system and an explanation system, required for all expert systems, together with a graphical output system, a geographical data input system and a geographical data description system required specifically for a cartographic expert system. The geographical descriptions, or metadata, could be incorporated into the main knowledge base, or stored independently as illustrated.

Software

There are two options in developing an expert system: use an existing inference engine, i.e. use a proprietary expert system shell; or develop an inference engine specifically for the system. An evaluation of several expert system shells indicated that none had the facilities required to develop a cartographic design expert system. The main difficulties are in integrating the inference mechanism with the required cartographic database and the

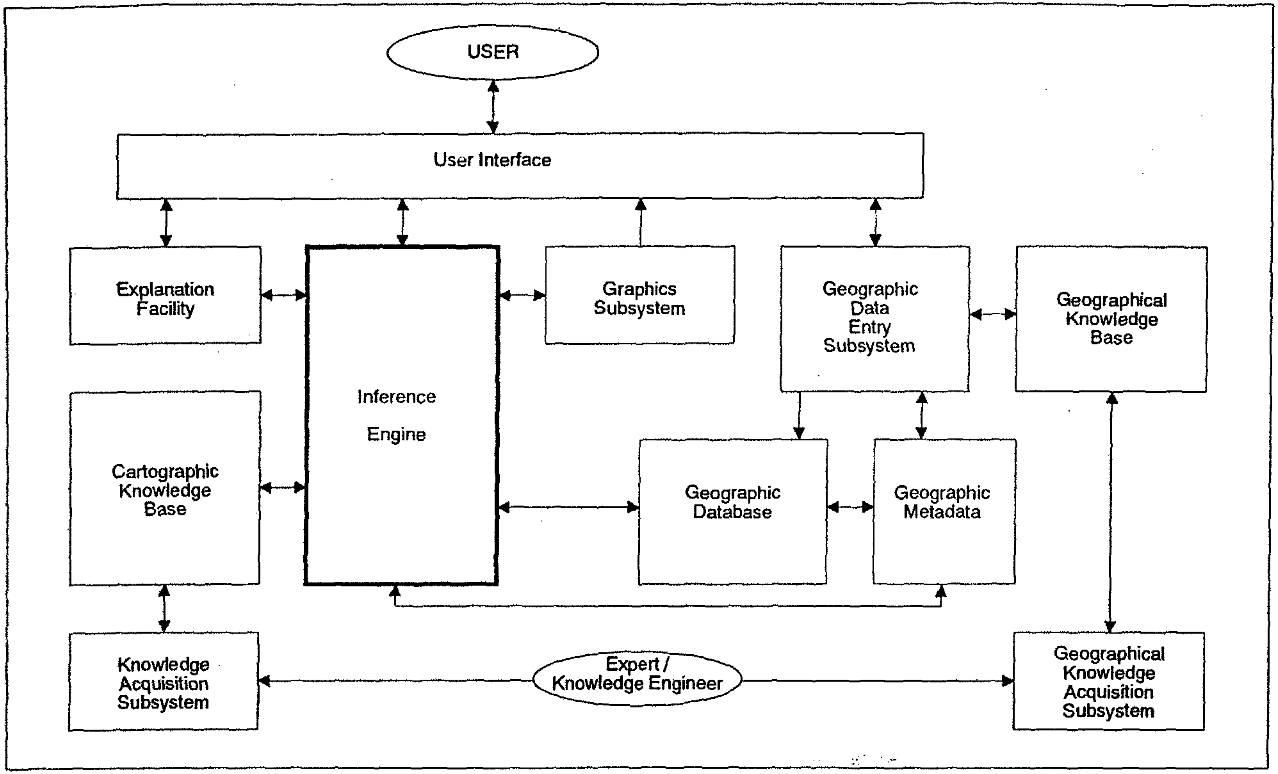

Figure 1. The Architecture of a Cartographic Design Expert System

production of graphic displays. Some shells, such as VP Expert and Knowledge Pro do have the ability to incorporate graphics, but these are either pre-created stored images or simple charts and diagrams. Thus, at an early stage a decision was taken to develop a customised inference mechanism. This view has also received support by others developing map design systems (e.g. [5, 6]). Knowledge engineering is frequently cited as being a major bottleneck in the development of expert systems. A customised inference mechanism has the advantage that the format of the knowledge base can be designed to facilitate the knowledge engineering process and the nature of the problem to be solved. Thus, the structures used to represent the knowledge can be as close as possible to those used by the domain expert or at least those that offer the most appropriate representation. This further supports the development of a customised inference engine.

Having decided to develop a specialist inference engine, the choice of development language must be considered. While traditional (procedural) languages such as Pascal, C and Basic have been used for some expert system development, they generally are viewed as being less than ideal and most opt to use 'fifth generation' languages. These languages support the declarative programming approach which is more suited to the type of problem solving involved with AI and ES rather than the procedural approach adopted by more traditional languages. Amongst those developing cartographic expert systems and systems in other fields, Prolog is one of the most popular languages and has been adopted here. Unlike conventional programs in languages like Pascal, in Prolog a description of the problem is given by a series of facts and rules and the program asked to find solutions to the problem. The essential premise is that the programmer describes what must be done, but the Prolog system itself organises how the computation is carried out. This is known as a 'declarative' approach, rather than the 'procedural' approach to programming adopted by Basic, Pascal, etc. where the programmer is also concerned with the mechanisms of solving the problem. This in theory should result in Prolog programs being shorter (i.e. requiring less lines of code) than procedural programs, easier to program, and easier to read and understand. While this may hold true for relatively simple programs, as program complexity increases programming in Prolog is not necessarily as simple of some of the introductory textbooks might imply. What is clear is that the thought processes involved in programming in Prolog are quite different from procedural programming and the conversion from programming in one paradigm to the other is not trivial.

The Inference Mechanism

The inference method adopted is a predominantly forward chaining mechanism, i.e. it is a data driven solution, where rules are matched against existing facts to establish new facts. Backward chaining, using hypothesis testing, is used at some points, mainly to establish if a default value is the most appropriate or to make a choice between options. An overall backward chaining approach has been attempted for map design (e.g.[5]), but in the author's opinion the lack of adequate evaluation procedures for assessing map design makes this approach less desirable. Merritt [4] points out that where it is not possible to enumerate all of the possible answers beforehand and have the system select the correct one, a goal driven, or backward chaining, approach will not work. This is because the variability of the inputs and the number of ways they can be combined is almost infinite.

The choice of inference method has some effect on the interface with the user. In a data driven system a number of facts must be initially established for the system to infer new facts from, whereas in a goal driven system facts are only gathered as they are needed. Thus, the initial fact gathering part of the mechanism may appear more like a conventional procedural program, although if correctly programmed only relevant questions will be asked.

Knowledge Representation

Frames are used as the main knowledge representation method for the map design specification. These are particularly appropriate for problems where we can identify a number of stereotypes, in this case a relatively small number of basic map themes and representation methods. A frame consists of title or name to identify it and a series of 'slots' which can contain a value. Each of the slots or blanks in the frame must be filled in order. This is achieved by a procedure or procedures being associated with each slot which may, for example, ask the user to answer a question, or refer to other frames. The resulting system has hierarchical structure where the topmost frames represent generalities and the lower ones are customised for more specific instances. Frame based systems have been identified as being particularly appropriate representations of objects in the design process [3] and have been suggested for knowledge representation in Cartographic Expert Systems by several authors.

The structure of frame based systems adds intelligence to the representation by allowing objects to inherit values from other objects. "Furthermore, each of the attributes can have associated with it procedures (called demons) which are executed when the attribute is asked for, or updated " [4]. In some examples, the inheritance feature of frames is used to reduce the number of slots in an individual frame by using a slot labelled 'a kind of' which indicates that the frame inherits the properties of the frame(s) further up the hierarchy. Slots for these properties are not included in the frame unless the value is different from the default. While this is a more compact representation than fully representing each frame, it is less clear. An alternative is to distinguish between frame definitions and instances of frames. Frame definitions specify the slots for a frame and instances provide specific values for the slots. Frame instances will be updated in working storage and accessed by the rules. For example, **Map** would be a frame definition and **Topographic** would be an instance of Map.

Frame: MAP		
SLOT	**FILLER**	**PROCEDURES**
Title	Optional	Ask_user
Author	*System user*	Ask_user
Date	*System date*	Ask_system
Map_topic	Compulsory	Build_menu
Map_class	Derived from topic	get_class
Notes	Optional	Ask_user
Description		
Map_user	*General*	Menu; if_change_rules
Map_purpose	*Overview*	Menu; if_change_rules
Output_media	*Screen*	Menu; if_change_rules
Level_of_detail	Computed	get_LOD
Layout		
Location	Compulsory	Build_Menu; if_change_rules
Format	*Fill screen*	Menu; when_added_rules; if_change_rules
Scale	*Max possible*	Menu; when_added_rules; if_change_rules
Selection		
Selection_index	Computed	compute_SI; if_change_rules
Base_information	List derived	Selection_rules
Thematic_information	List derived	Selection_rules
Symbol_specification		
Specifications	List derived	Fill_representation_frames

Notes - Filler column - *italics* = default value.

Figure 2. The MAP frame

Examples of the MAP frame are given in Figures 2 and 3. The generic MAP frame (Fig. 2) indicates the slots to be filled, the kind of information that will fill them and the procedures that are used to fill them, or that come into operation when the value in the slot is altered. A frame from the second level of the hierarchy is illustrated in Fig. 3a. There are three possible frames at this level, for 'basic', 'cultural' and 'physical' maps which exist largely as a starting point for designing maps of topics 'unknown' to the system. At the third level of the hierarchy frames represent individual map topics, and at the fourth and most detailed level, frames represent individual map designs, which have all slots filled (Fig. 3b). Space does not allow the members of lists or details of the procedures to be shown in these diagrams, but a complete frame, or map specification for a topographic map of Nigeria is shown in Figure 4.

There is also a series of cartographic representation frames. Each of these contains slots for the parameters specifying the symbols in enough detail for them to be drawn. These slots are filled by default values, procedures, or by reference to look-up tables containing colour sequences, point symbols, line patterns, etc.

The rules associated with the slots are represented in the system in a number of ways. Those procedures which are basic to the system and would apply universally to any cartographic design system, such as those for calculating and checking scale and format, are integrated with the inference mechanism. Others more specific to this system are separated into a number of modules to allow for easy modification.

As the size of the knowledge base grows, performance becomes problematic. Various indexing systems can be used to speed up the process required to find the next rule to solve. The rules in forward chaining systems generally need to be indexed by more complex methods than required for backward chaining systems. By using the MAP frame definition as the controlling structure for inference and filling the slots in a specific order to create instance frames, complex rule indexing methods may be avoided initially. They may be required if full flexibility is to be achieved at the Modify stage of design. The problem is also reduced in MapDesigner by declaring a number of different rule and fact structures for specific aspects of the problem, rather than trying to use generic rule and fact structures to cover all situations. This is essential as the problems to be solved for different slots are unique to that stage in the process. The whole nature of the problem, and hence the solution, is quite different to the 'classical' classification type of expert system typically described.

Although the order of processing the slots is fixed in the inference engine, any slot may in fact have its value pre-assigned, i.e. the system can load partially complete frames. Before calling the procedures associated with a slot, a check is made to see if it is already filled. This allows, for example, certain standard characteristics to be retained for a series of maps or to maintain a 'house' style.

The User Interface

The user interface to the system is menu based. The user is never confronted with a blank screen and left wondering what to do next. Two basic screen arrangements are used, one during text based queries and the other when graphics are being displayed. This is due to the simpler handling of textual information with PDC Prolog in text mode rather than graphics mode. Responses to menus are entered via the keyboard. Wherever possible default values are given to questions (both with and without menus) so the user only needs to press the F10 or enter key to accept that response. It is important that the menus shown and the default values highlighted are not simply pre-programmed into the system. Many of the menus are generated by the system at run time by extracting appropriate information from the knowledge base. A good example of this is the Map Topic menu. This is different to most other menus in that a tree or hierarchical menu is constructed.

As can be seen from the MAP frame definitions (Fig. 2), map types (topics or themes) are defined hierarchically with each topic belonging to a higher level class. The information stored in the knowledge base includes a set of facts with these two values specified, representing the set of map topics the system 'knows' about. At run time the system uses map_topic and map_class to build a tree menu as shown in Figure 5. The further to the right in this menu the user can select the type of map required, the more assistance can be offered by the system. A less well defined map topic will require more interaction with the user.

Frame:	(a) MAP: Basic	(b) MAP: Geological
SLOT	FILLER	FILLER
Title	Optional	**Nigeria Geology**
Author	*System user*	**David Forrest**
Date	*System date*	**01/01/95**
Map_topic	**Basic**	**Geological**
Map_class	**Basic**	**Physical**
Notes	Optional	
Description		
Map_user	*General*	**Specialist**
Map_purpose	*Overview*	**Overview**
Output_media	*Screen*	**Screen**
Level_of_detail	Computed	**4**
Layout		
Location	Compulsory	**(2,3,15,15)**
Format	*Fill screen*	**Fill screen**
Scale	*Max possible*	**7500000**
Selection		
Selection_index	Computed	**7**
Base_information	*known list of options*	**[known list]**
Thematic_information	*known list of options*	**[known list]**
Symbol_specification		
Specifications	List derived	**[List of symbols]**

Notes - example columns- *italics* = default value, **bold** = slot filled, [ ] = list, details omitted.

Figure 3 a & b. Examples of MAP frames.

```
fmap_date(dated(1995,4,27))
fmap_author("D. Forrest")
fmap_title("Nigeria - Topographic")
fmap_type("topographic")
fmap_purpose("overview")
fmap_user("author")
foutput_media("screen")
flevel_of_detail(4)
fscale(7500000)
fformat(18,16.5)
flat_long(2.5,15,3.5,14)
flimits(277.98731656,388.9402453,1667.9238994,1541.2843966)
fselect_index(9)
fbase_info_list(["Coastline","Seas","Major Rivers","Lakes","Lake_fill",
"International Boundaries","Capitals","Main Highways","Main Relief"])
fbase_info("Main Relief",10)
fbase_info("Main Highways",10)
fbase_info("Capitals",10)
fbase_info("International Boundaries",10)
fbase_info("Lakes",10)
fbase_info("Major Rivers",10)
fbase_info("Coastline",10)
ftheme_info_list([])
frepresentation("Coastline","boundaries - one level")
frepresentation("Seas","isolated areas")
frepresentation("Rivers","network - branching")
frepresentation("Lakes","boundaries - one level")
frepresentation("Lake_fill","isolated areas")
frepresentation("Administrative Boundaries","boundaries - hierarchy")
frepresentation("Settlements","ranked points")
frepresentation("Roads","network - link & node")
frepresentation("Relief","hypsometric layers")
fsymbolism("Coastline","Coastline",symbolspec("line","blue","dark",
"low","continuous","",0,"fine",201))
fsymbolism("Seas","Seas",symbolspec("area","cyan","light","low",
"solid","",0,"",158))
fsymbolism("Rivers","Major Rivers",symbolspec("line","blue","dark",
"mid", "continuous","",0,"fine",190))
fsymbolism("Administrative Boundaries","International Boundaries",
symbolspec("line","grey","dark","low","chain","chain",0,"thick",210))
fsymbolism("Lakes","Lakes",symbolspec("line","blue","dark","mid",
"continuous","",0,"fine",202))
fsymbolism("Lake_fill","Lake_fill",symbolspec("area","cyan","light",
"low","solid","",0,"",191))
fsymbolism("Settlements","Capitals",symbolspec("point","magenta",
"dark","mid","geometric","square",0,"medium",251))
fsymbolism("Roads","Main
Highways",symbolspec("line","red","mid","mid","cased","main_highways",
0,"thick",223))
fsymbolism("Relief","0-500",symbolspec("area","green","mid","low",
"solid","",0,"",141))
fsymbolism("Relief","500-2000",symbolspec("area","brown","mid","low",
"solid","",0,"",141))
fsymbolism("Relief","over 2000",symbolspec("area","brown","mid","mid",
"solid","",0,"",141))
```

Figure 4. Example map specification frame.

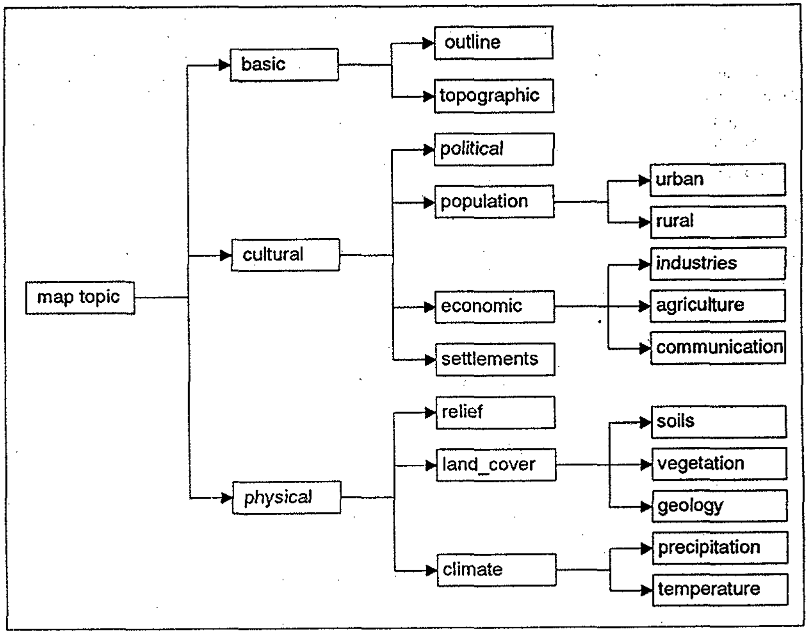

Figure 5. Map Topic menu

DATABASE & METADATA

The database

Although the system is capable of producing a specification for an abstract map, in order to produce an actual map it is first necessary to have some information to map. In the longer term it is desirable that the system be capable of interfacing with a number of databases, but it is expedient to limit the interface to a customised database in the first instance. Database files read by the system are all of one of four formats: node files which contain point information; chain files which contain line data; polygon files which contain the boundaries of areas; and data files which contain attribute information about points or areas.

Metadata

There are some differences in the use of the term metadata in current literature on GIS. In some cases it is used to refer to rather general, global characteristics of databases or data sets, containing information about sources, availability, dataset quality, etc., to enable potential users to evaluate their utility before purchasing or otherwise acquiring the data. The alternative use refers to a more detailed description of individual elements of the dataset and is perhaps more an extension to the data dictionary incorporated into most database management systems. Although there is obviously overlap between these two levels of knowledge, the differences are significant. Here the term is used to refer to the more specific information describing in some detail every type of element in the database.

A long term goal is to develop an expert system capable of interactively building this metadata or knowledge base about the data, shown in Figure 1 as the Geographical knowledge acquisition sub-system. Currently the knowledge has been determined manually based upon analysis of the data using the guidelines about phenomena and information. As the system developed, the importance this metadata became increasingly obvious, although it is rarely if ever discussed in developing design expert systems, where the emphasis is usually on representational issues. Having determined the metadata characteristics these are entered into a series of facts in a knowledge base called 'Meta'. The expert system to be developed to acquire this knowledge would quite likely use a frame based model as again there are stereotypical situations, and the storage structure used by 'Meta' would allow this approach.

Four different types of metadata are stored in the metadata knowledge base. These are the main information about classes of data, details of individual co-ordinate files, details about data (attribute) files and lookup tables to translate between names for features used in the system's knowledge base and their names in the database. These are referred to as meta_data, coord_file, data_file and look_up respectively. The structure and possible contents of the first and most significant of these which contains the primary description of the data is illustrated in Figure 6. Further details of the how this structure was developed and explanation of the possible values can be found in Forrest [1] and in more depth in Forrest [2].

CONCLUSIONS

This paper has outlined the basic structure of the MapDesigner system and its major components. Tests of the system show that it is capable of producing a wide variety of types of map over a range of scales from 1:2million to 1:15million, the specific limits set during development. The maps produced change in their content and level of detail depending upon the scale selected and their intended use. The system is currently limited to producing maps for screen display. A considerable development of the knowledge base will be required to match the requirements of other output media, but the same basic principles and structures will apply. The success of the system indicates that as long as reasonable goals are set it is possible to develop cartographic design expert systems which at least produce maps which conform to good cartographic practice.

Frame: META_DATA	
SLOT	Possible values ( ) = comment
feature category	(symbolic name for feature class used in MapDesigner expert system)
data capture date	
comment	
phenomenon	discrete (point at map scale) linear specific areas continuous surface
spatial data	points lines boundaries (polygons not explicit) polygons cells
attribute level	identical feature coded hierarchical_feature_coded ordinal interval ratio_absolute ratio_density ratio_derived external (not defined here)
nature of phenomenon	tangible conceptual
symbolic name of co-ordinate file	(reference to 'coord_file' not actual name)
lookup file name	(actual file name)
symbolic name of data file	(reference to 'data_file')
symbolic name in data file	(the name of the attribute or column)

Figure 6. Meta_data slots and possible values.

References

[1] Forrest, D. (1991) Classifying cartographic representations for cartographic design expert systems. *Proceedings, 15th International Cartographic Conference.*

[2] Forrest, D. (1995) The application of expert systems to small scale map design. *Unpublished PhD dissertation,* University of Glasgow.

[3] Lansdown, R.J. (1987) Graphics, Design and Expert Systems. In: Earnshaw, R.J. (ed.) **Theoretical Foundations of Computer Graphics and CAD.** Berlin: Springer-Verlag. pp.1153-1174.

[4] Merrit, D. (1989) **Building Expert Systems in Prolog.** New York: Springer-Verlag.

[5] Muller, J-C., Wang, Z. (1990) A Knowledge Based System for Cartographic Symbol Design. *Cartographic Jounal.* Vol.27, No.1, pp.24-30.

[6] Zhen, T., Wang, J.Y., Lin, A.S. (1993) Knowledge representation and reasoning in nautical chart design. *Proceedings, 16th International Cartographic Conference.*